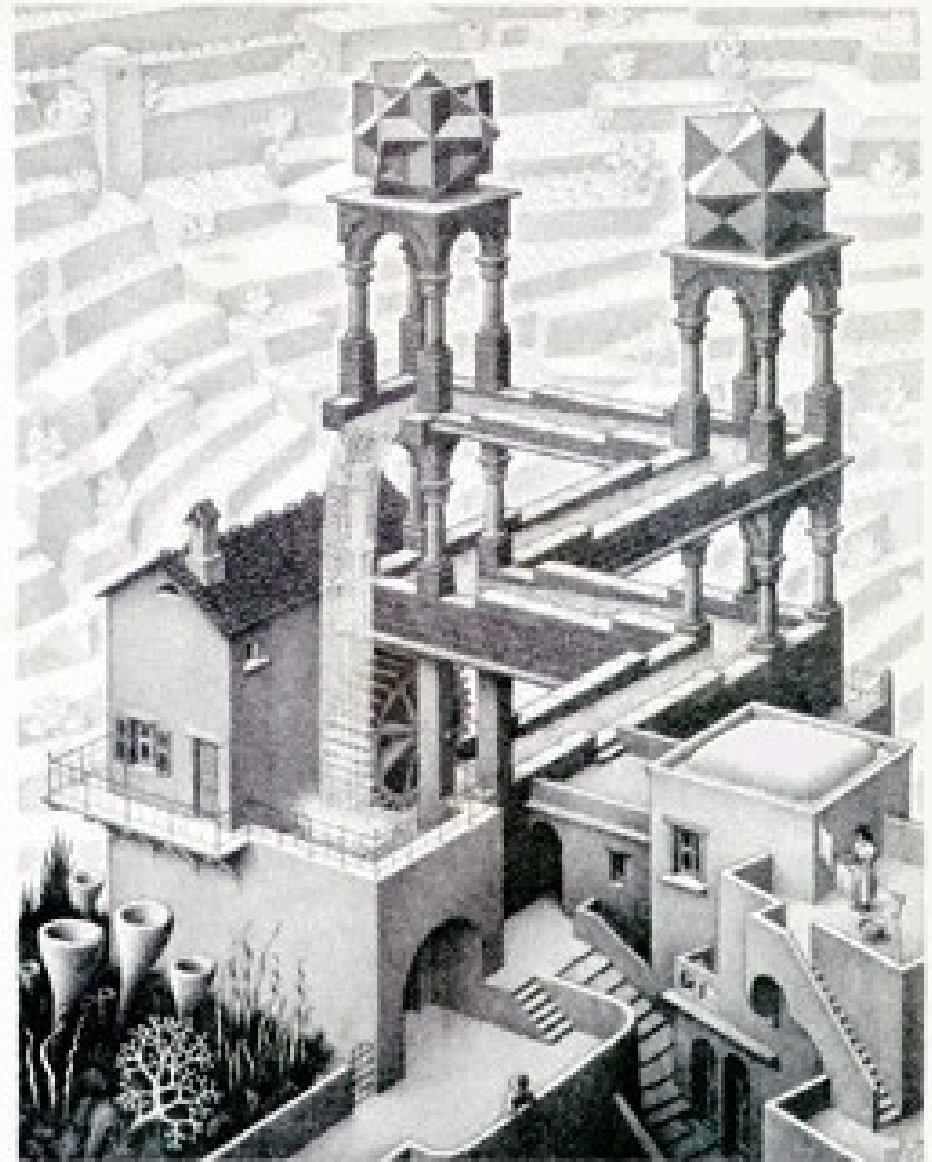


Agile Project Management

Is the waterfall eternal?



van Spaandonk & Koks

Meet John van Spaandonk



- PhD electronics, 2 years business admin.
- Project / partner management
*embedded software in Copiers & Mobile com.,
Public transport, R&D, internet technologie*
- “Waterfall”, SCRUM, RUP, Prince2
- Consultancy
EFQM, CMM, Business Balanced Scorecard
- Training & coaching
Project management, personal development
- International
US, UK, Germany, Sweden, Poland, Belgium
<http://www.vanspaandonk-koks.nl>

Presentation Outline

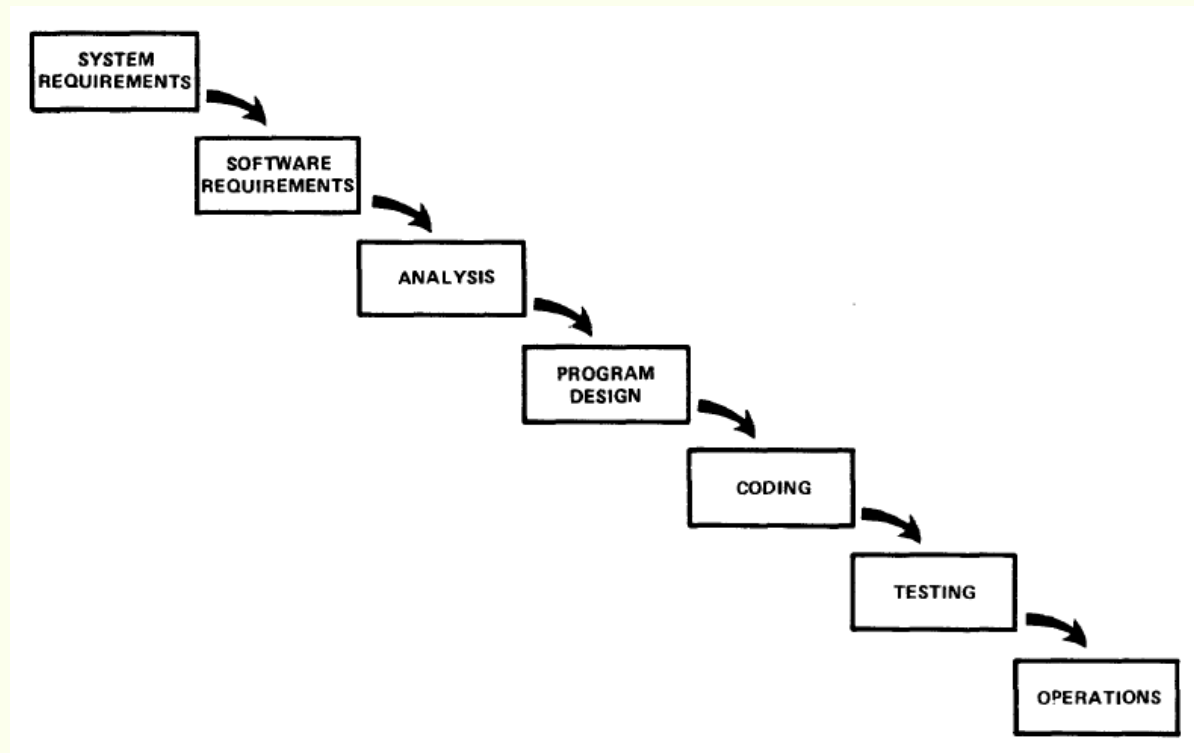
- Winston Royce never recommend a simple “one-pass sequential” waterfall model
- Tens of billions of dollars were lost in US defence contracting, partly due to use of waterfall model
- Iterative methods were suggested by US defence over waterfall model since at least 20 years
- Agile development is not just a further development of the waterfall model, it dates from before WWII (1930's)
- Agile development has advantages in a real world environment
- Applicability of agile depends upon specific circumstances

Question

- What is your experience with the “waterfall” model?
- Are you familiar with the original 1970 paper by Winston Royce?

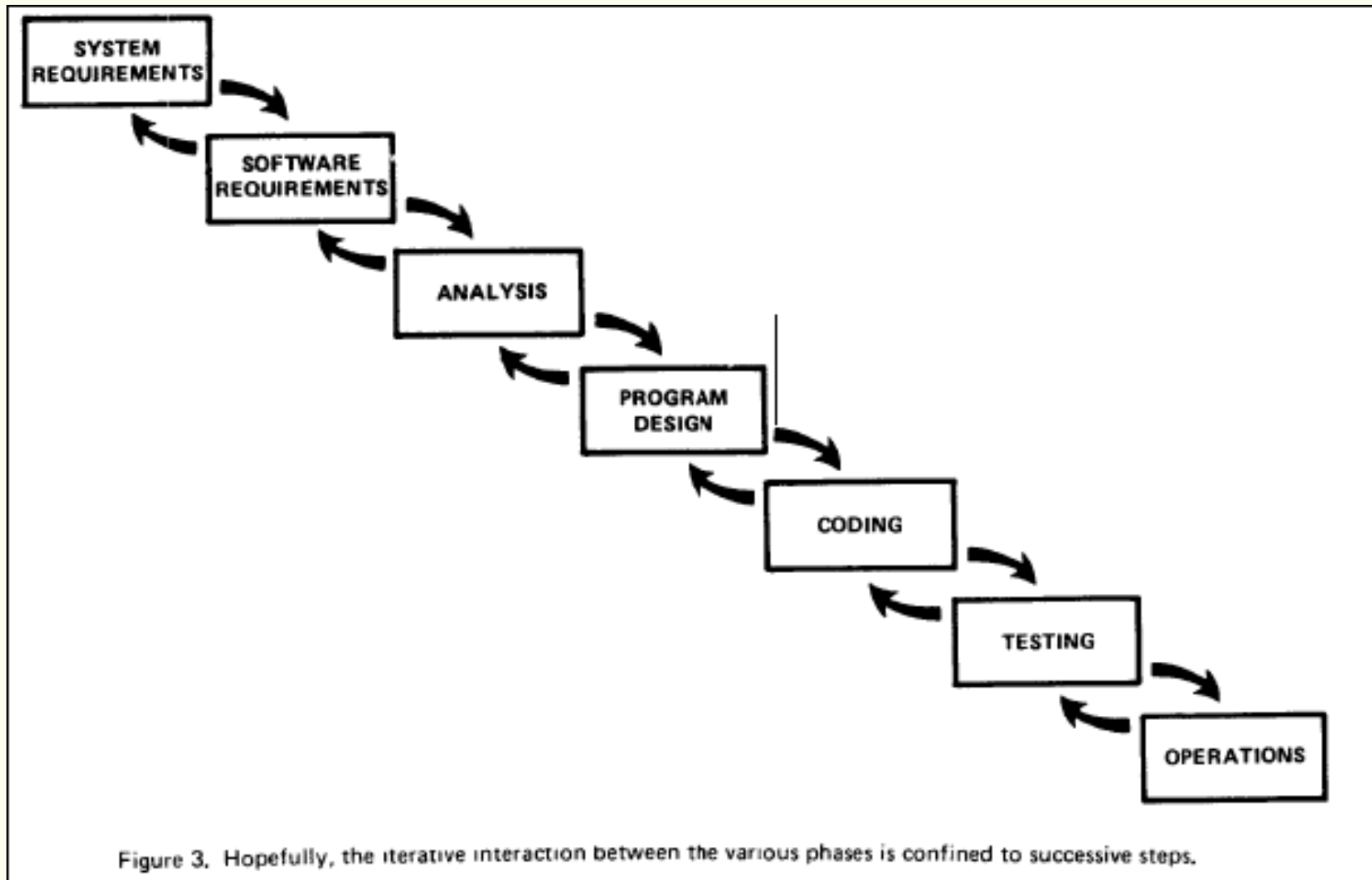
1970: Winston W. Royce

“Managing the development of large software systems”

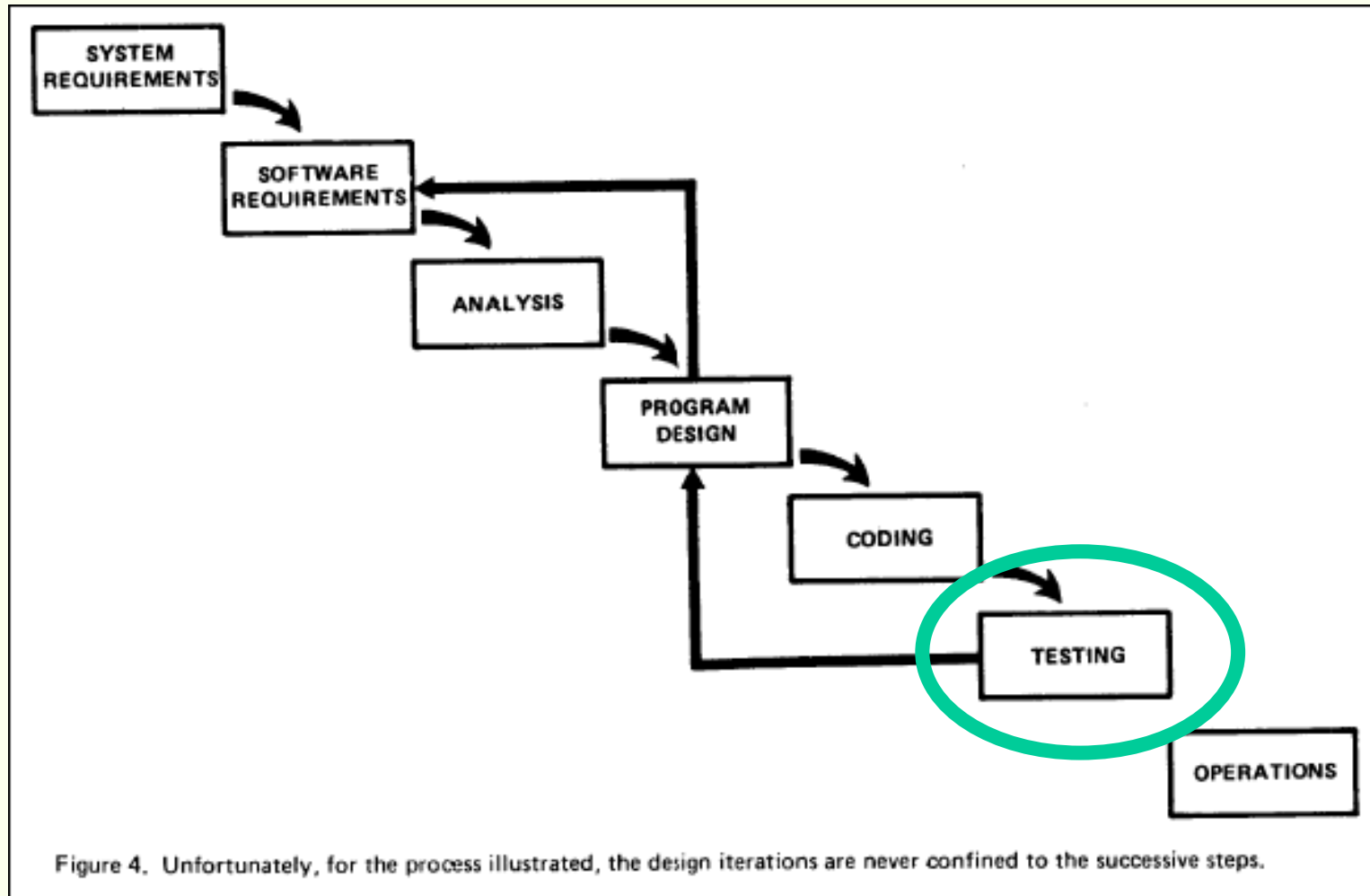


“I believe in this concept, **but the implementation described above is risky and invites failure**”

Why does Royce say this?



Feedback is global!



- You find out too late what the problems are...

Modifications of the simple model

- Program design comes first
- Document the design
- Do it twice
- Plan, control and monitor testing
- Involve the customer

Necessary modifications

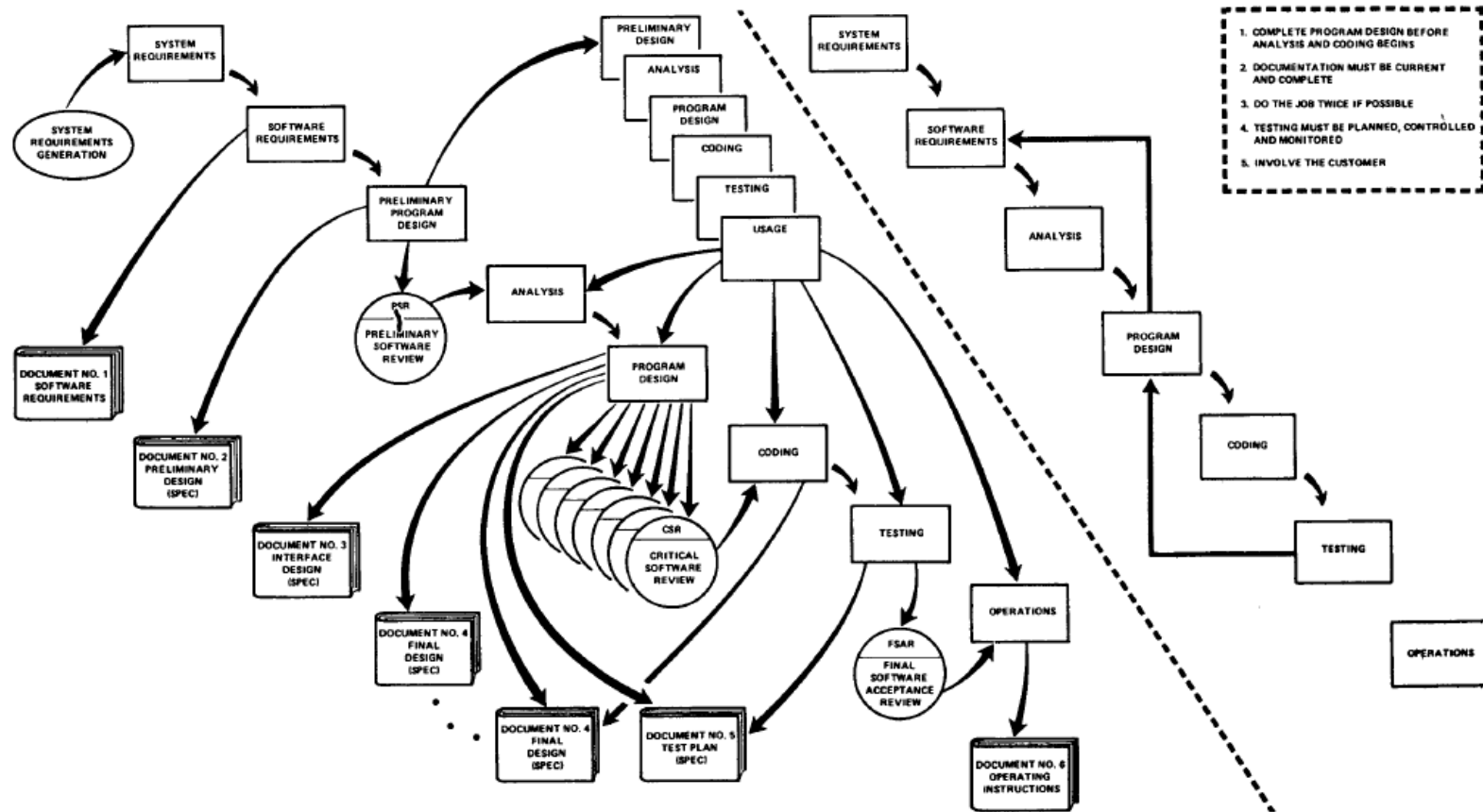


Figure 10. Summary

Royce's advice is not heeded...

- Royce intended to demonstrate that the one-pass model does not work (note that he did not coin the name “waterfall”)
- However, pure one-pass waterfall model was used by DOD for subcontracting:
 - Create specification
 - Get quotations on basis of specification
 - Select lowest priced proposal
 - Accept results at the end
- DoD-Std-2167

Fast forward to 1987: Frederic P. Brooks

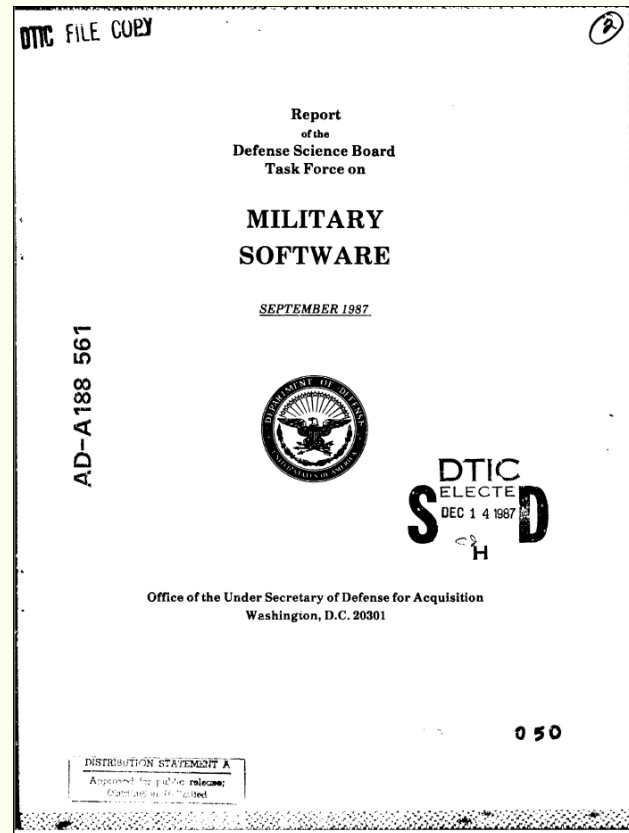
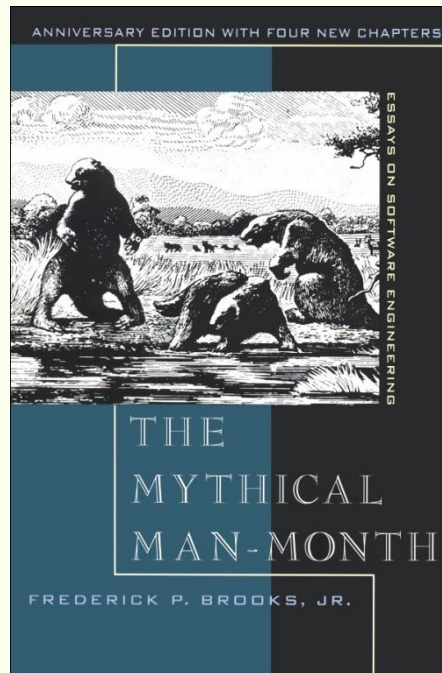


- Managed development of OS/360
- Wrote about it in “The mythical man month”.

Brooks’ law:

“Adding manpower to a late software project makes it later”

- **Chairman Defense Science Board on Military Software**



21 mei 2007

van Spaandonk & Koks

Brooks' verbatim recommendations

- “DoD-Std-2167 likewise needs a radical overhaul to reflect modern best practice. Draft 2167A is a step, but it does not go nearly far enough.
- As drafted, it continues to reinforce exactly the document-driven, specify-then-build approach that lies at the heart of so many DoD software problems....
- In the decade since the waterfall model was developed, our discipline has come to recognize that [development] requires iteration between the designers and users.”

Summary of problems with waterfall

- Project is divided in fixed number of development stages
- Early commitment
- Little room for changes
- Not suitable if requirements are poorly understood or likely to change during project

Agile

van Dale:

- beweeglijk
- vlug
- levendig
- alert
- wakker
- waakzaam



The agile manifesto

- ***Individuals and interactions***
- ***Working Software***
- ***Cooperation with stakeholders***
- ***React to changes***

More important than

- Processes and tools
- Comprehensive documentation
- Contract negotiations
- Following a detailed plan

www.agilealliance.org

Characteristics of Agile Development

- Prioritized requirements (e.g. MoSCoW)
- Frequent interaction with customers, marketing, ...
- Iterative time boxing with feedback
- Releasable product after each iteration
- Projects built around trusted individuals
- Focus on face-to-face communication
- Adaptive control (not static)
- Suitable for unpredictable situations
- Complemented by project mgt system like Prince2

Agile versus “pure” Waterfall

Waterfall

One pass

Predictive

Fixed, pre-planned dev.

Focus on Artifacts

Agile

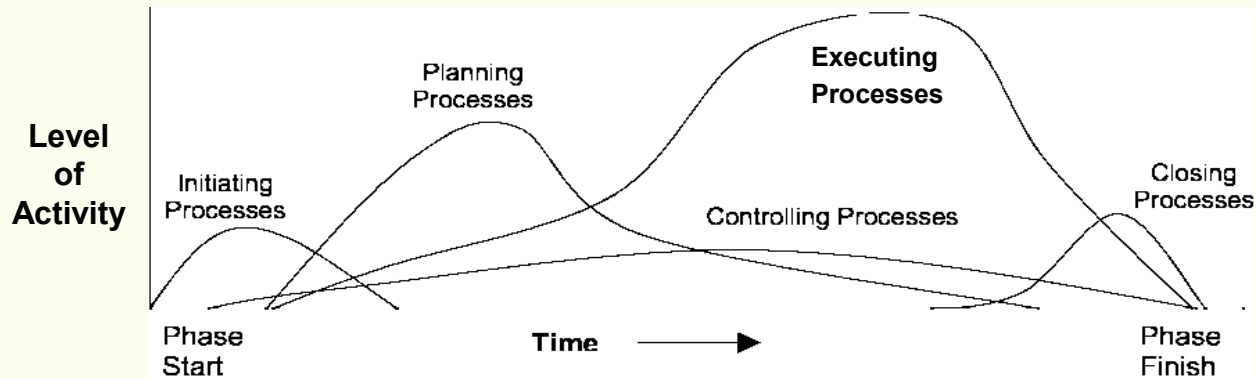
Iterative

Adaptive

React to
changes

Focus on
tested features

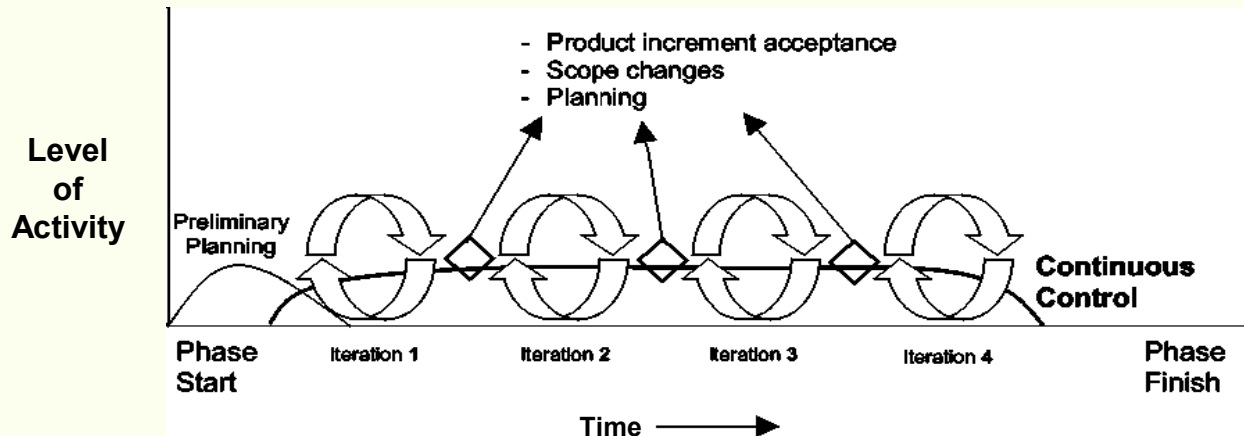
Traditional



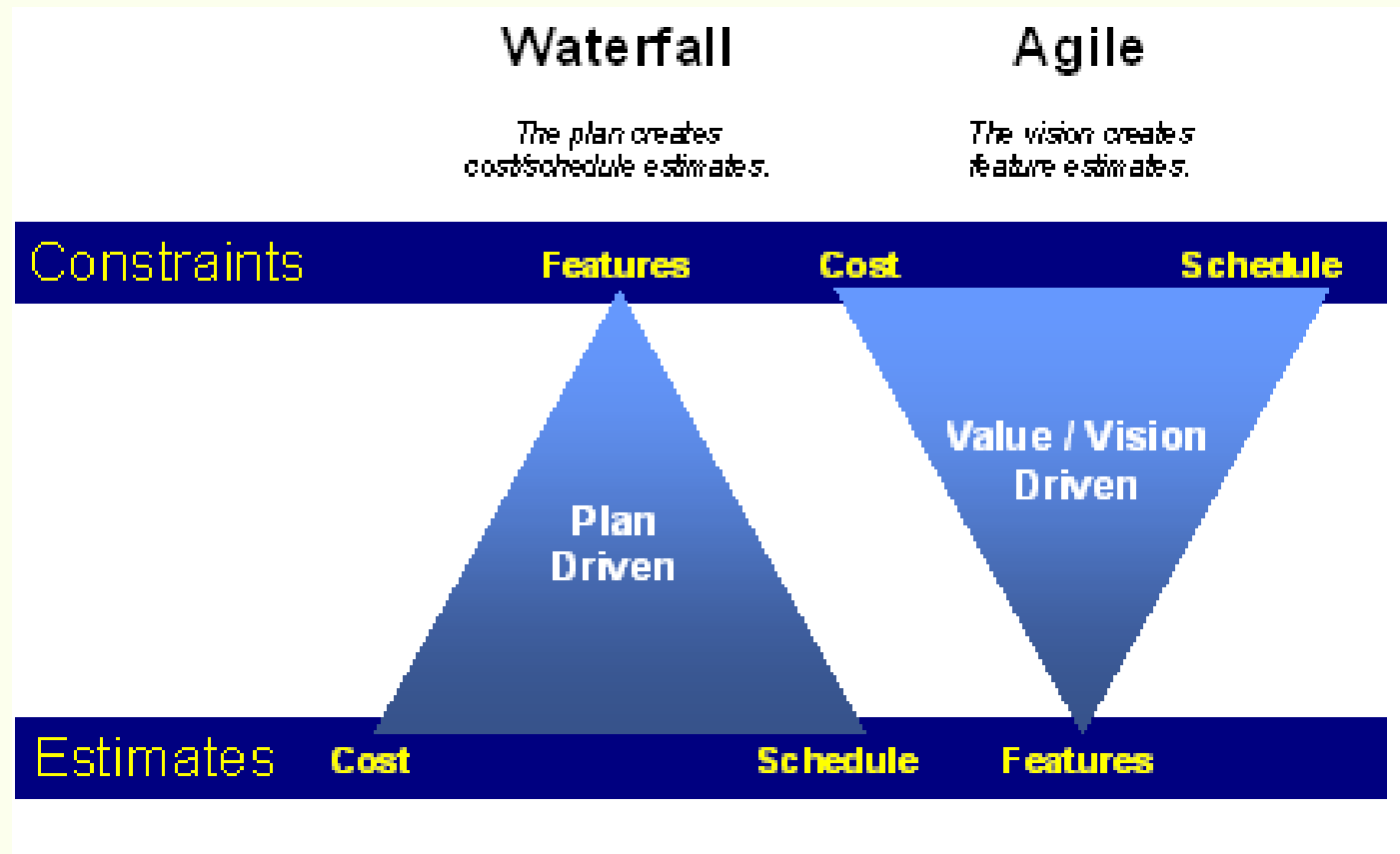
From PMBOK® Guide process groups in a phase

Agile

Iteration Contains:
Fixed Scope
Planning
Execution



Agile reverts the “devils triangle”



Agile versus iterative development

Agile:

- increments + feedback mechanism (customer involvement after each iteration)
- focus on working software necessary for feedback mechanism to work
- timebox measured in weeks, not months
- product scope can change after each iteration
- focus on collaboration and interaction

Agile is applicable when...

- Organization culture must support negotiation
- People must be trusted
- Fewer but more competent people
- Accept decisions made by developers
- Environment must support rapid communication between team members
- Works very well for small, co-located teams
- Perfect for Internet projects
Create prototype, gather feedback, implement more screens (iterations of 1 to 2 weeks possible)

Agile is not so easily applicable when...

- Large organization with many teams
- Junior developers
- Different development sites
- Command-and-control culture
- Requirements fixed / non negotiable
- Subcontracting with “standard” fixed-price

Agile and CMM

- Mark Paulk (SEI) on Agile and CMM:
 - CMM describes what not how
 - Agile is set of best practices, some with narrow application domain
 - Agile may adress many CMM level 2 and 2 practices
 - Implement CMM in spirit of agile
- Mike Konrad and James Over (SEI) on Agile and CMMI
 - CMMI for the Organization, TSP for the IT Shop

Agile versus TSP

Agility vs. the Team Software Process <i>Agile software development is anything but antithetical to TSP.</i>	
Agile Value Statement	How TSP Relates
<i>Individuals and interactions over processes and tools</i>	<p>TSP holds that the individual is key to product quality and effective member interactions are necessary to the team's success.</p> <ul style="list-style-type: none"> ◆ Project launches strive to create gelled teams. ◆ Weekly meetings and communication are essential to sustain them. ◆ Teams define their own processes in the launch.
<i>Working software over comprehensive documentation</i>	<p>TSP teams can choose evolutionary or iterative lifecycle models to deliver early functionality—the focus is on high quality from the start. TSP does not require heavy documentation.</p> <ul style="list-style-type: none"> ◆ Documentation should merely be sufficient to facilitate effective reviews and information sharing.
<i>Customer collaboration over contract negotiation</i>	<p>Learning what the customer wants is a key focus of the launch. Sustaining customer contact is one reason for having a customer interface manager on the team.</p> <ul style="list-style-type: none"> ◆ Focus on negotiation of a contract is more a factor of the organization than of whether TSP is used.
<i>Responding to change over following a plan</i>	<p>TSP teams expect and plan for change by:</p> <ul style="list-style-type: none"> ◆ Adjusting the team's process through process improvement proposals and weekly meetings. ◆ Periodically relaunching and replanning whenever the plan is no longer a useful guide. ◆ Adding new tasks as they are discovered; removing tasks that are no longer needed. ◆ Dynamically rebalancing the team workload as required to finish faster. ◆ Actively identifying and managing risks.

Discussion

- Is agile for you?